**PART 2: Programming Answers**
Please check the below programs to verify answers

- apt321_ss11381_ML_Homework_1.ipynb (Python Notebook File)
- apt321_ss11381_ML_Homework_1.py (Python Executable Script)

1. **What was the estimated value of P(C) for C = 1?**
   P(Spam)= 0.4018006002

2. **What was the estimated value of P(C) for C = 0?**
   P(Not Spam)= 0.5981993998

3. **What were the estimated values for (ˆμ, ˆσ 2) for the Gaussian corresponding to attribute capital run length longest and class 1 (Spam).**
   Mean for capital run length longest and class 1= 97.2091286307
   Variance for capital run length longest and class 1= 36369.9911126

4. **What were the estimated values for (ˆμ, ˆσ 2) for the Gaussian corresponding to attribute char freq ; and Class 0.**
   Mean for char_freq_; and class 0= 0.0484258639911
   Variance for char_freq_; and class 0= 0.0883056032571

5. **Which classes were predicted for the first 5 examples in the test set?**
   [0, 0, 0, 0, 0]

6. **Which classes were predicted for the last 5 examples in the test set?**
   [0, 0, 0, 0, 0]

7. **What was the percentage error on the examples in the test file?**
   No. of Correct Predictions = 160
   No. of Incorrect Predictions = 40
   Percentage Error = 20.0

8. **Sometimes a not-very-intelligent learning algorithm can achieve high accuracy on a particular learning task simply because the task is easy. To check for this, you can compare the performance of your algorithm to the performance of some very simple algorithms. One such algorithm just predicts the majority class (the class that is most frequent in the training set). This algorithm is sometimes called Zero-R. It can achieve high accuracy in a 2-class problem if the dataset is very imbalanced (i.e., if the fraction of examples in one class is much larger than the fraction of examples in the other). What accuracy is attained is you use Zero-R instead of Gaussian Naive Bayes?**

   The accuracy of Zero-R Classifier is ~59% (Precise= 0.58999999999999997) which is low as compared to performance of Naive-Bayes Classifier.

**P.T.O**

**9. Gaussian Naive Bayes is based on two assumptions: (1) the conditional independence assumption, and (2) the assumption that the pdfs for p(xj |C) are Gaussian. These assumptions are more reasonable for some datasets than for others. Do you think these assumptions are reasonable for the spam dataset you just used? Why or why not? In answering this question, you can give a common-sense argument and/or show relevant plots, graphs, or statistical information. (Note that Gaussian Naive Bayes can sometimes be effective even if the assumptions are not very reasonable. In order to do correct classification, it is enough to determine the correct MAP class. It is not necessary to actually compute the correct posterior probability P(C|x) for each class.)**

A Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

1. Consider a scenario where we have to identify whether a fruit is an apple or not.
Some parameters may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as 'Naive'.

2. Similarly, we can consider this scenario applying to our prediction whether the email is "Spam" or "Not Spam".
Features like "charfreq;", "charfreq(", "charfreq[", "charfreq!", "charfreq$", "charfreq#" are independent of each other as each field represents count of different characters respectively.
Features such as "capital_run_length_average", "capital_run_length_longest", "capital_run_length_total" are also independent from each other as they are completely dependent on changes in input data(word).
Also, when assumption of independence holds, a Naive Bayes classifier performs better compare to other models like Zero-R Classifier.

Therefore, we can say that the existence of other features are not dependent on each other and we can classify this problem using Naive Bayes.